

Homework 1: Finding Similar Items: Textually Similar Documents

[Submit Assignment](#)

Due Nov 9 by 11:59pm **Points** 5 **Submitting** a file upload

The homework can be done in a group of 2 students.

Submission on time, i.e. before the deadline, is awarded with 3 exam bonus points if your homework is accepted, i.e. you have successfully presented and demonstrated your homework to a course teaching assistant during a presentation session.

Task

You are to implement the stages of finding textually similar documents based on Jaccard similarity using the *shingling*, *minhashing*, and *locality-sensitive hashing* (LSH) techniques and corresponding algorithms. The implementation can be done using any big data processing framework, such as Apache Spark, Apache Flink, or no framework, e.g., in Java, Python, etc. To test and evaluate your implementation, write a program that uses your implementation to find similar documents in a corpus of 5-10 or more documents such as web pages or emails.

The stages should be implemented as a collection of classes, modules, functions or procedures depending the framework and the language of your choice. Below, we give a description of sample classes that implement different stages of finding textually similar documents. You do not have to develop the exact same classes and data types as described below. Feel free to use data structures that suit you best.

1. A class *Shingling* that constructs k -shingles of a given length k (e.g., 10) from a given document, computes a hash value for each unique shingle, and represents the document in the form of an ordered set of its hashed k -shingles.
2. A class *CompareSets* that computes the Jaccard similarity of two sets of integers – two sets of hashed shingles.
3. A class *MinHashing* that builds a minHash signature (in the form of a vector or a set) of a given length n from a given set of integers (a set of hashed shingles).
4. A class *CompareSignatures* that estimates similarity of two integer vectors – minhash signatures – as a fraction of components, in which they agree.



5. (**Optional task for extra 2 bonus**) A class *LSH* that implements the LSH technique: given a collection of minhash signatures (integer vectors) and a similarity threshold t , the *LSH* class (using banding and hashing) finds all candidate pairs of signatures that agree on at least fraction t of their components.

To test and evaluate scalability (the execution time versus the size of input dataset) of your implementation, write a program that uses your classes to find similar documents in a corpus of 5-10 documents. Choose a similarity threshold s (e.g., 0,8) that states that two documents are similar if the Jaccard similarity of their shingle sets is at least s .

Datasets

- For documents, see the datasets in [the UC Irvine Machine Learning Repository](https://archive.ics.uci.edu/ml/index.php) (<https://archive.ics.uci.edu/ml/index.php>), or find other documents such as web pages or emails.
- To find more datasets follow [this link](https://github.com/caesar0301/awesome-public-datasets) (<https://github.com/caesar0301/awesome-public-datasets>)

Readings

- [Finding Similar Items.pptx](#)  | [Finding Similar Items.pdf](#) 
- [Chapter 3 \(https://www-cambridge-org.focus.lib.kth.se/core/services/aop-cambridge-core/content/view/FDDA225F039792324F28D73D26950E89/9781139924801c3_p68-122_CBO.pdf/finding-similar-items.pdf\)](https://www.cambridge-org.focus.lib.kth.se/core/services/aop-cambridge-core/content/view/FDDA225F039792324F28D73D26950E89/9781139924801c3_p68-122_CBO.pdf/finding-similar-items.pdf) in *Mining of Massive Datasets*, by Jure Leskovec, Anand Rajaraman, and Jeffrey D. Ullman, 2nd edition, Cambridge University Press, 2014

Submission, Presentation and Demonstration

To submit your homework, you upload your solution in a zip file to Canvas. Canvas records the submission time. **Submission on time, i.e. before the deadline, is awarded with 3 exam bonus points if your homework is accepted.** Bonus will not be given, if you miss the deadline. **Your homework solution must include**

1. Source code if required (with comments);
2. Makefile or scripts to build and run;
3. Report (in PDF) with a short description of your solution, instructions how to build and to run, command-line parameters, if any (including default values), results, e.g., plots or screenshots.

Within a week after the homework deadline, **you present and demonstrate** your homework on your own laptop to a course instructor. A Doodle pool will be provided to **book a time slot for presentation**.

Grading and Bonus Policy

The grade for a homework is **pass/fail**. If you submit report your homework on time and your solution is accepted, you will get **3 bonus points** on your first ID2222 exam whenever you take it. Some homework include **optional tasks for extra bonus**.